

# Competitive Air Hockey Database

Alexandria Duell and Rita Berglund  
House of Pucks

## Overview

Air Hockey is a relatively new competitive sport and is played competitively (or at least has a competitive player living in the state) in 11 states, with the main three being North Carolina, Texas, and Illinois. State tournaments can range from 30-90+ competitors. The world tournaments have on average 90-100+ competitors. There are also weekly tournaments which take place mostly in the main three states and have between 10-40 competitors. Leagues are less common but are formatted as a weekly event that lasts about 2 months per season and usually have between 10-15 competitors.

Many players compete in multiple matches, sets, and games per event. This produces thousands of individual game records annually. Manual spreadsheets are not sufficient as they cannot reliably manage hundreds of players and games per season, thousands of games annually, and historical match and ranking data. Currently, Air Hockey uses multiple third-party apps, which can be confusing for both experts and novices of the sport. It creates unnecessary time upkeep and misinformation on players and tournaments.

A database system is needed to: store player profiles and regional affiliations; record match, set, and game results; support multiple competition formats; enable accurate statistics, rankings, and historical queries; and scale as participation and events grow. One source of truth would help to grow the sport and streamline recording tournaments and matches. It would also allow for novices to become more involved as they will be able to use the app with ease. In the database, we are planning to initially record player information, matches, sets, games, and locations of tables. The database will be designed to be able to expand upon the initial implementation to record tournaments, leagues, and compute statistics.

## Database Outline

### **locations**

Records the physical venues where air hockey tables are located, matches can be held, and/or tournament locations. This entity tracks both public + private properties.

#### **Attributes**

`location_id`: int, auto\_increment, unique, not NULL, PK  
`owned_by_player_id`: int, nullable, FK references `players`  
`num_of_tables`: int, not NULL, default 0  
`street_address_1`: varchar, not NULL  
`street_address_2`: varchar, nullable  
`city`: varchar, not NULL  
`state`: varchar, not NULL  
`country`: varchar, not NULL, default 'USA'  
`zip_code`: int, not NULL  
`type_of_address`: enum ('residential', 'commercial', 'club', 'bar', 'other', etc.), not NULL  
`location_name`: varchar, nullable  
`notes`: varchar, nullable

#### **Constraints**

- The combination of `location_name`, `street_address_1`, `street_address_2`, `city`, `state`, `country`, + `zip_code` must be unique

#### **Relationships**

- M:1 relationship with `players`, implemented with `player_id` via `owned_by_player_id` as a FK in `locations`
- 1:M relationship with `matches`, implemented with `location_id` as FK in `matches`

## players

Records all people who participate in or are involved with USAA Air Hockey competitions, including officials and hosts. This is essential for tracking player information, contact details, + statistics.

### Attributes

```
player_id: int, auto_increment, unique, not NULL, PK
first_name: varchar, not NULL
last_name: varchar, not NULL
gender: enum ('M', 'F', 'Other', 'Prefer not to say'), nullable
primary_email: varchar, unique, nullable
primary_phone_num: varchar, unique, nullable
street_address_1: varchar, nullable
street_address_2: varchar, nullable
city: varchar, nullable
state: varchar, nullable
country: varchar, not NULL, default 'USA'
zip_code: int, nullable
```

### Constraints

- `primary_email` has a unique constraint when not NULL

### Relationships

- 1:M relationship with `locations`, implemented with `owned_by_player_id` as FK in `locations`
- M:N relationship with `matches` implemented through `player_matches` junction table
- 1:M relationship with `matchOfficials`, implemented with `player_id` via `official_player_id` as FK in `matchOfficials`
- 1:M relationship with `matches`, implemented with `player_id` via `winner_id` as FK in `matches`
- 1:M relationship with `sets`, implemented with `player_id` via `winner_id` as FK in `sets`
- 1:M relationship with `games`, implemented with `player_id` via `winner_id` as FK in `games`

## matches

Records all competitive Air Hockey matches between 2 players. This is the central entity for tracking match information including timing, location, + outcome. A match consists of a max of 3, 5, or 7 sets. The first player to win the required 2 of 3, 3 of 5, or 5 of 7 sets wins the match.

### Attributes

```
match_id: int, auto_increment, unique, not NULL, PK
location_id: int, not NULL, FK references locations
winner_id: int, nullable, FK references players
max_sets: int, not NULL, default 3, check (max_sets in (3, 5, 7))
faceoff_type: enum ('standard', 'puck flip'), not NULL, default 'standard'
start_datetime: datetime, not NULL
end_datetime: datetime, nullable
match_type: enum ('challenge', 'tournament', 'league', etc.), not NULL
match_status: enum ('scheduled', 'in_progress', 'completed', 'abandoned'), not NULL, default 'scheduled'
notes: varchar, nullable
```

### Constraints

- `end_datetime` must be greater than `start_datetime` when both are not NULL
- `winner_id` must be NULL unless `match_status` = 'completed'

- `max_sets` can only be 3, 5, or 7

## Relationships

- M:1 relationship with `locations`, implemented with `location_id` as FK in `matches`
- 1:M relationship with `sets`, implemented with `match_id` as FK in `sets`
- M:N relationship with `players`, implemented with `player_matches` junction table
- 1:M relationship with `matchOfficials`, implemented with `match_id` as FK in `matchOfficials`
- M:1 relationship with `players`, implemented with `player_id` via `winner_id` as FK in `matches`

## `player_matches`

Junction table that implements the many-to-many relationship between `matches` + `players`. This entity records all matches per each individual player.

### Attributes

- `player_match_id`: int, auto\_increment, unique, not NULL, PK
- `player_id`: int, not NULL, FK references `players`
- `match_id`: int, not NULL, FK references `matches`
- `starting_side`: enum ('left', 'right'), not NULL
- `is_winner`: boolean, not NULL, default FALSE

### Constraints

- Unique constraint on combined `player_id` + `match_id`
- Exactly 2 `player_matches` records must exist per match
- Only one player can have `is_winner` = TRUE per match

## Relationships

- M:1 relationship with `players`, implemented with `player_id` as FK in `player_matches`
- M:1 relationship with `matches`, implemented with `match_id` as FK in `player_matches`
- Implements the M:N relationship between `players` and `matches`

## `sets`

Records individual sets within a match. A set consists of multiple games until a player wins 4 games out of that set. A set contains 4-7 games.

### Attributes

- `set_id`: int auto\_increment, unique, not NULL, PK
- `match_id`: int, not NULL, FK references `matches`
- `winner_id`: int, nullable, FK references `players`
- `set_num`: int, not NULL, check (`set_num` between 1 and 7)
- `start_datetime`: datetime, nullable
- `end_datetime`: datetime, nullable
- `set_status`: enum ('scheduled', 'in\_progress', 'completed', 'abandoned'), not NULL, default 'scheduled'

### Constraints

- Unique constraint on `match_id` + `set_num` - set numbers must be unique within a match
- `end_datetime` must be greater than `start_datetime` when both are not NULL
- `winner_id` must be NULL unless `set_status` = 'completed'
  - A set must contain at least 4 games when completed
- `set_num` cannot exceed the parent match's `max_sets` value

## **Relationships**

- M:1 relationship with `matches`, implemented with `match_id` as FK in `sets`
- 1:M relationship with `games`, implemented with `set_id` as FK in `games`
- M:1 relationship with `players`, implemented with `player_id` via `winner_id` as FK in `sets`
- 1:M relationship with `matchOfficials`, implemented with `set_id` as FK in `matchOfficials`

## **games**

Records individual games within a set. A game is played until 1 player reaches 7 points. There are at least 4 games in a set, and a max of 7 games.

### **Attributes**

`game_id`: int, auto\_increment, unique, not NULL, PK  
`set_id`: int, not NULL, FK references `sets`  
`player_1_id`: int, not NULL, FK references `players`  
`player_2_id`: int, not NULL, FK references `players`  
`winner_id`: int, nullable, FK references `players`  
`player_1_pts`: int, not NULL, default 0, check (`player_1_pts` between 0 and 7)  
`player_2_pts`: int, not NULL, default 0, check (`player_2_pts` between 0 and 7)  
`game_num`: int, not NULL, check (`game_num` between 1 and 7)  
`start_datetime`: datetime, nullable  
`end_datetime`: datetime, nullable  
`game_status`: enum ('scheduled', 'in\_progress', 'completed', 'abandoned'), not NULL, default 'scheduled'

### **Constraints**

- Unique constraint on `set_id + game_num` - game numbers must be unique within a set
- `player_1_id + player_2_id` must be different (a player cannot play against themselves)
- `player_1_id + player_2_id` must match the 2 players in the parent match
- `winner_id` must be NULL unless `game_status` = 'completed'
- `winner_id` must be either `player_1_id` or `player_2_id` when not NULL
- `end_datetime` must be greater than `start_datetime` when both are not NULL
- When `game_status` = 'completed', one player must have exactly 7 points

## **Relationships**

- M:1 relationship with `sets`, implemented with `set_id` as FK in `games`
- M:1 relationship with `players`, implemented with `player_id` via `player_1_id` as FK in `games`
- M:1 relationship `players`, implemented with `player_id` via `player_2_id` as FK in `games`
- M:1 relationship with `players`, implemented with `player_id` via `winner_id` as FK in `games`

## **matchOfficials**

Records officials (referees + witnesses) for matches and/or sets. Records who officiated, in what capacity, and for what type of entity (`matches` and/or `sets`).

### **Attributes**

`matchOfficial_id`: int, auto\_increment, unique, not NULL, PK  
`officialPlayer_id`: int, not NULL, FK references `players`  
`match_id`: int, nullable, FK references `matches`  
`set_id`: int, nullable, FK references `sets`  
`official_type`: enum ('referee', 'witness'), not NULL  
`sets_reffed`: int, not NULL, default 1

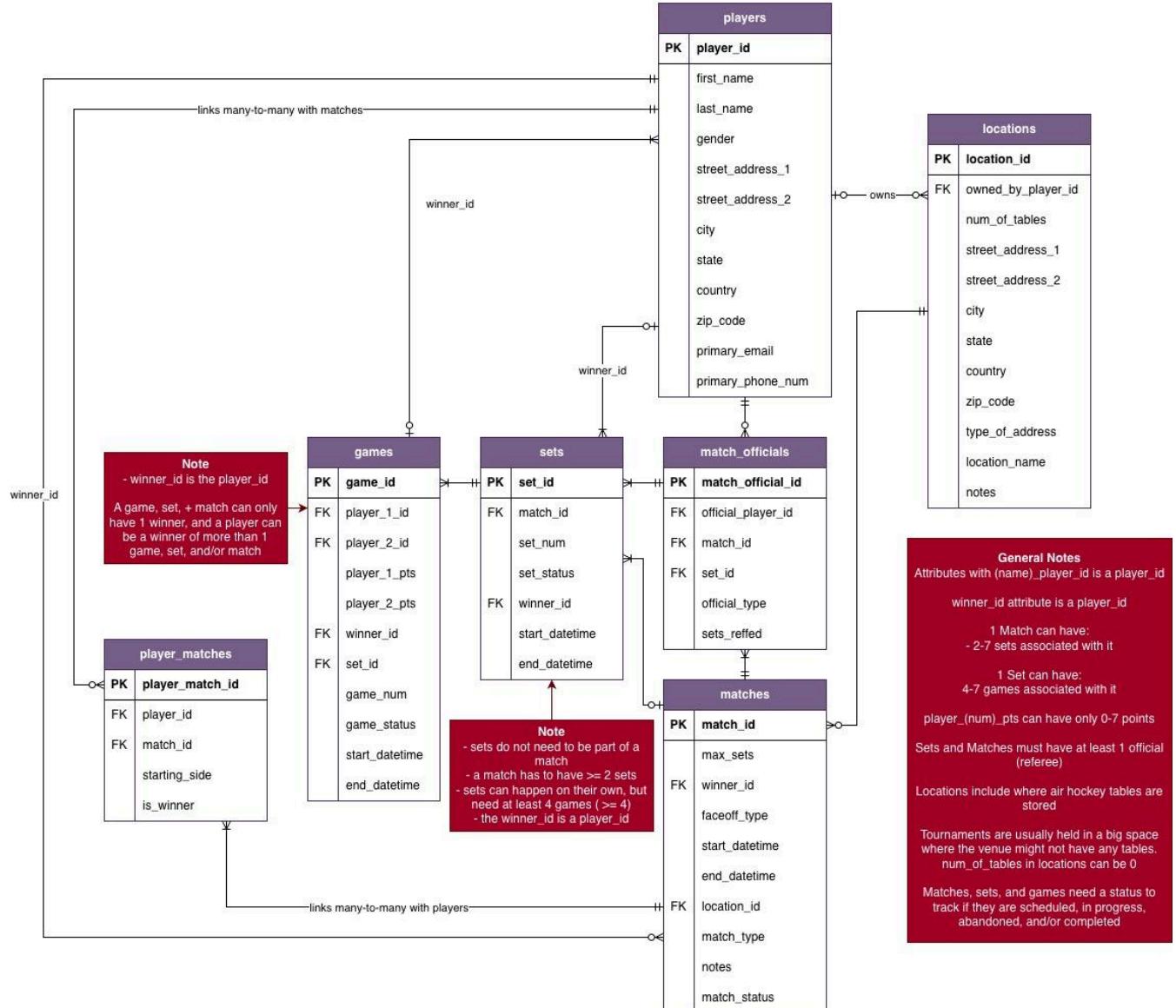
## Constraints

- If `match_id` is not NULL, then `set_id` must not be NULL - if there is a `set_id`, then `match_id` can be NULL
- `official_player_id` cannot be 1 of the players competing in the match
- `sets_reffed` should not exceed the total number of sets in the match

## Relationships

- M:1 relationship with `players`, implemented with `player_id` via `official_player_id` as FK in `matchOfficials`
- M:1 relationship with `matches`, implemented with `match_id` as FK in `matchOfficials`
- M:1 relationship with `sets`, implemented with `set_id` as FK in `matchOfficials`

## Entity-Relationship Diagram



## Citations

All content in this document for the database design is original work from Alex Duell + Rita Berglund. We did research + referenced the USAA Rules + Regulations for challenge matches + games via the NC Air Hockey Players Association website: FAQs. NC Air Hockey Players. (2025). <https://www.ncairhockeyplayers.com/faqs>. Used documents found in the section "Is there a Rulebook for Air Hockey?"